

TITLE OF THE INVENTION
High Performance Transparent Call Distribution

5 CROSS REFERENCE TO RELATED APPLICATIONS

 This application claims priority under 35 U.S.C. §119(e) of U.S. Provisional Patent Application No. 60/437,937 filed January 3, 2003.

10 STATEMENT REGARDING FEDERALLY SPONSORED RESEARCH OR
 DEVELOPMENT

--Not Applicable--

BACKGROUND OF THE INVENTION

15 The invention pertains to the field of initiating calls or sessions in communications networks.

 In systems providing communications services, it is known to distribute requests for services among a plurality of service providing entities, such as servers or individual processors
20 within a server. The individual servers or processors perform the bulk of the processing and data transfer associated with the sessions, and the overall system is designed to achieve a desired balance of cost and performance for a desired capacity. However, one performance bottleneck that has remained in such systems is
25 the entity responsible for assigning the service requests among the servers or processors. In some cases, the volume of signaling traffic or "session dialogs" associated with a large number of service requests can slow down the performance of a system even when there are sufficient processing resources for handling the
30 intra-session activities of all requested sessions.

 To address the performance issues associated with handling session dialogs, it has been known to utilize so-called "proxy servers" in systems having a distributed service architecture.

For example, in systems utilizing the Session Initiation Protocol (SIP), a SIP proxy server contains information identifying alternative servers that are available to handle service requests. For each request, the proxy server selects a server to handle the request, re-formats the request message from the requestor, and sends the re-formatted request to the selected server. All subsequent traffic between the requestor and selected server flows through the proxy server. Although this arrangement has the benefit that all requestors see only a single point of service, the proxy server(s) can easily become bottlenecks that reduce performance. Additionally, the proxy server(s) add to overall system cost.

Another mechanism employed in certain systems, including SIP-based systems, is the use of a "redirect" message. Using such a message, the recipient of a session initiation message explicitly signals a session initiator to redirect its session initiation request to another server. While such redirection has the benefit that the initial recipient retains no state information about the request nor is involved in subsequent session data transfers, it unfortunately causes a significant increase (substantially double) in session signaling traffic. This increase in signaling traffic can lead to undesirable communications and processing bottlenecks.

It would be desirable to provide for the distribution of session initiation requests among a plurality of servers and/or processors in a manner that is transparent to session initiators and that provides for high throughput of signaling traffic in a scalable fashion.

BRIEF SUMMARY OF THE INVENTION

In accordance with the present invention, a media server system is disclosed in which calls or session requests are assigned to processors in a manner that is substantially

transparent to the requestor and that provides for high-throughput processing of session initiation dialogs.

5 The media server system includes a dispatcher and a collection of processors, which may reside in the same physical system as the dispatcher or in a separate physical system. The sources of session initiation dialog send initial session requests using a well-known port identifier associated with the dispatcher. In one embodiment, the sources of session initiation dialogs include application servers that provide high-level communications services such as conferencing. Upon receiving each initial message, the dispatcher selects one of the processors to conduct the session initiation dialog and forwards the initial message to the selected processor. The forwarding of the initial message may be accomplished by modifying the port number of the message to a port number uniquely associated with the selected processor, and then sending the message into a switch fabric or other interconnection that routes the message to the processor associated with the modified port number.

20 Upon receiving an initial message of a session initiation dialog from the dispatcher, each processor creates a response message including the port identifier uniquely associated with the processor to identify the port to which subsequent messages of the session initiation dialog are to be directed. The response message is then sent to the application server that was the source of the initial message. Subsequent messages of the session initiation dialog can be sent directly to the selected processor by the application server, bypassing the dispatcher. Thus, the bulk of session initiation dialog is handled in a distributed fashion by the same processors that control the media services associated with session, substantially enhancing performance while maintaining the desired degree of transparency obtained using a centralized dispatcher.

Other aspects, features, and advantages of the present invention will be apparent from the Detailed Description that follows.

5 BRIEF DESCRIPTION OF THE SEVERAL VIEWS OF THE DRAWINGS

Figure 1 is a block diagram of a communications network with a media server in accordance with the present invention;

Figure 2 is a block diagram of a media server in the system of Figure 1; and

10 Figure 3 is a signaling diagram illustrating certain signaling operations of the system of Figure 1, including internal signaling in the media server of Figure 2.

DETAILED DESCRIPTION OF THE INVENTION

15 The disclosure of U.S. Provisional Patent Application No. 60/437,937 filed January 3, 2003 is hereby incorporated by reference.

In Figure 1, a communications system includes a plurality of application servers (AS) 10 and a media server (MS) 12 that
20 provide communications services such as conferencing and interactive voice response (IVR) functions, among other services. These services are provided to a plurality of media endpoints (ME) 18 which may include media gateways, voice over IP (VOIP) phones, personal computers or similar devices. Media
25 streams of communications sessions are carried on media connections 14, which are created, modified, and terminated via signaling transactions on separate signaling channels 16. The signaling transactions are in accordance with a session management protocol such as Session Initiation Protocol (SIP) or
30 Media Gateway Control Protocol (MGCP). The signaling transactions are delivered over the network via transport protocols such as User Datagram Protocol (UDP). The media

streams are delivered in accordance with a media transport protocol such as the Real Time Protocol (RTP).

For economic reasons, it is desirable that the MS 12 be capable of providing media services for a large number of concurrent sessions with minimal service latency, i.e., minimal delay between the receipt of a request for service and the delivery of a media stream. For handling a large number of concurrent media streams, it has been known to provide multiple independent processors within a media server and some means for assigning media streams to the different processors. As the media-processing capacity of the MS 12 increases, however, there is a commensurate increase in the amount of associated signaling traffic for managing the media connections 14. A high volume of signaling traffic can cause increased service latency and/or undue limitations on the number of concurrent media streams that are allowed.

Figure 2 shows an arrangement for a MS 12 that provides for high performance handling of a relatively large volume of signaling traffic so as to minimize the impact of the signaling traffic on MS performance. Incoming SIP signaling messages, shown as "ingress traffic" in Figure 2, are received by an ingress processing function 20. As described in more detail below, the signaling messages are either forwarded to a SIP dispatcher 22 or to one of several application processors (AP) 24 via a switch fabric 26. The SIP dispatcher 22 is also capable of forwarding messages to the APs 24 via the switch fabric 26.

The general problem addressed by the presently disclosed technique is to utilize the plurality of APs 24 to achieve high overall performance of the MS 12. As already mentioned, it has been known to assign different SIP sessions to different APs 24 so as to enable the MS 12 to carry a large number of concurrently active media sessions. It would be possible to assign unique SIP addresses (IP addresses) to the various APs 24 such that each AP

24 can be directly accessed from an external device such as an AS 10 (Figure 1). However, such an arrangement would be undesirable from a system perspective. In such a case, it would potentially be necessary to configure each AS 10 with all of the unique SIP addresses of the various APs 24 as well as information describing the kinds of traffic to be directed to each one. In large scale systems, the number of IP addresses required to support such an arrangement becomes prohibitive. Furthermore, whenever the MS 12 were to be upgraded or re-configured in a manner affecting the number or capabilities of the APs 24 or their respective SIP addresses, it would be necessary to also re-configure each AS 10 so as to maintain a coherent view of the MS 12 throughout the system. Such complexities at the system level are preferably avoided.

Therefore, the MS 12 is assigned one SIP address that becomes well-known to all the ASs 10 through configuration information. Additionally, the MS 12 utilizes a well-known port identifier, such as port number 5060, to which initial SIP requests are to be directed. The various APs 24 are each assigned a unique port identifier different from the well-known port identifier. The port identifier of an AP 24 assigned to a session is sent to a requesting AS 10 during the initial part of the session signaling, in the manner described below. It may be convenient to assign port identifiers to APs 24 in some predetermined order such as 5061, 5062, . . ., or to utilize another suitable port identifier assignment scheme.

The operation of the system of Figures 1 and 2 is illustrated with respect to an initial portion of a SIP session as shown in Figure 3. An AS 10 desiring to initiate a media session between an ME 18 and the MS 12 creates a SIP INVITE message and sends it to the well-known SIP address and port number of the MS 12. If the well-known address is 10.1.1.1 and the well-known port

number is 5060, for example, then the Request line of the INVITE message will look as follows:

INVITE sip:svc@10.1.1.1:5060 SIP/2.0

5 where "10.1.1.1:5060" is the Request Uniform Resource Indicator (URI) identifying the MS 12 as the target of the INVITE message, and where "svc" represents the name of a SIP service type offered by the MS 12. Such service types include announcements, conferencing, interactive voice response (IVR), and dialogs. As
10 is known in the art, the Request line is followed by various header lines and the body of the SIP message.

 The INVITE message is received by the ingress processing function 20 of the MS 12 (Figure 2), which uses classification/filter processing to identify the message as a SIP
15 INVITE. Once so identified, the message is removed from the normal ingress flow and forwarded to the SIP dispatcher 22 to be assigned to an AP 24.

 When the SIP dispatcher 22 receives the INVITE message, it selects an AP 24 to handle the SIP session. This selection may be
20 made in a variety of ways. It may be desirable that different APs 24 are optimized for different types of services, in which case part of the selection process is to determine the service type being requested and the identity of one or more APs 24 that are designated for the requested service type. Generally, there will
25 be multiple APs 24 available for a given service type, so that it will be necessary to perform further selection from among the available APs. This selection can be performed based on certain information in the SIP INVITE message. For example, a hash function can be applied to the Call Identifier or Conference
30 Identifier appearing in the INVITE message, and the output of the hash function is used to identify an AP 24. Other information from the SIP message may also be used, such as the "To" and "From" fields. The SIP message information that is used to select an AP

should be globally unique, so that messages belonging to the same session are assigned to the same AP 24. This ensures that in the case of re-transmissions, only one AP 24 receives all the traffic for a given session.

5 To optimize the assignment process, the dispatcher 22 may implement an additional communications channel between itself and the APs 24 (not shown in the Figures). Such a channel is used by the APs 24 to periodically update the dispatcher 22 with load information and to report when calls have been terminated. This
10 enables the dispatcher 22 to make assignments based on up-to-date loading information, resulting in high overall utilization of the MS 12.

Effectively supporting conferencing services poses special challenges. Conferences are multi-party sessions, whereas other
15 services generally consume a single session. All requests for a specific conference must be sent to the same AP to eliminate the need for multiple AP's to coordinate and share conference data. Although it is possible to obtain load information for a particular conference by parsing the incoming signaling messages,
20 such an approach is expensive in terms of resource utilization. The additional processing load required for this message handling would negatively affect the performance and scalability of the MS 12. To solve this problem, a load hint in the form of a "size" parameter is provided in an easily accessible location in the
25 Request URI (e.g. sip:conf=1234@ms1.network.com; size=64). The size parameter may optionally be set by applications that wish to gain the best utilization of the MS 12. If the size parameter is present, the dispatcher 22 uses it in the process of selecting an AP 24. If the parameter is not present, the dispatcher 22 applies
30 a default load when making the assignment. In either case, if the feedback channel to the dispatcher 22 is present, the APs 24 update the dispatcher 22 with their actual load values.

After the SIP dispatcher 22 selects an AP 24, it changes the port number appearing in the UDP datagram to the port number of the selected AP 24. Thus, if the selected AP 24 has port number 5065 assigned to it, then the destination UDP port number of the resulting request datagram will be 5065. This modified INVITE message is then forwarded to the selected AP 24 via the switch fabric 26. In contrast to a SIP proxy, the dispatcher 22 does not modify message headers such as Via and/or Record Route. Only the port number in the request datagram is changed. The SIP dispatcher 22 does not monitor any subsequent messaging for, nor keep any SIP signaling state pertaining to, this session.

The selected AP 24 then performs the normal processing associated with a SIP INVITE request, which initially includes creating a response message shown as "200 OK" and sending this response message to the requesting AS 10. As shown in Figure 3, this response message is sent directly to the AS 10 bypassing the SIP dispatcher 22. The 200 OK message includes a Contact field that is used by the responding AP 24 to notify the requesting AS 10 to direct further messages directly to the responding AP 24 rather than to the well-known address and port number. This is accomplished by placing the port number of the responding AP in the Contact field, i.e.:

Contact: <sip:svc@10.1.1.1:5065>

In the example shown in Figure 3, the AS then performs a similar INVITE/200 OK sequence with the ME 18. The 200 OK messages are acknowledged by sending corresponding ACK messages to the MS 12 and ME 18. The ACK message back to the responding AP 24 uses the address obtained from the Contact field of the 200 OK message from the AP 24. Referring back to Figure 2, the ingress processing function 20 of the MS 12 receives the ACK message and forwards it to the correct AP 24 via the switch fabric 26 bypassing the SIP dispatcher 22. The forwarding of messages

within the MS 12 may be performed using conventional techniques employing circuit or flow identifiers that are assigned to messaging flows by internal switching control logic (not shown). Once the SIP dispatcher 22 has selected an AP 24 for a given Call or Conference ID, this information can be given to the internal switching control logic and to the ingress processor 20 to facilitate the routing of later-received messages directly to the correct AP 24.

Once the assigned AP 24 has received the ACK message, the AS 10 and AP 24 have exchanged sufficient information to create the media channel 14 for the session on behalf of an ME 18, which as shown in Figure 1 extends directly between the ME 18 and the MS 12. The media flow for the remainder of the session can be conducted in the normal manner between these two elements.

Although the illustrated embodiment employs a switch fabric 26 for routing media streams to/from the APs 24, it may be possible to employ other forms of interconnection in alternative embodiments, including a local-area network (LAN). Also, it may be desirable in alternative embodiments to separate the dispatcher 22 into a separate physical system from the APs 24, i.e., to employ a modified media server similar to that shown in Figure 2 but having a communications interface to a separate dispatcher system in place of the dispatcher 22.

It will be apparent to those skilled in the art that modifications to and variations of the disclosed methods and apparatus are possible without departing from the inventive concepts disclosed herein, and therefore the invention should not be viewed as limited except to the full scope and spirit of the appended claims.